

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-268659

(43)Date of publication of application : 20.09.2002

(51)Int.Cl.

G10L 13/00

G06F 3/16

G10L 13/08

G10L 13/06

(21)Application number : 2001-067258

(71)Applicant : YAMAHA CORP

(22)Date of filing : 09.03.2001

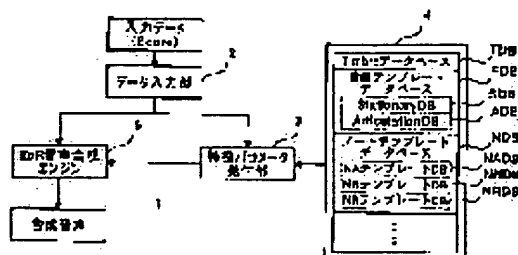
(72)Inventor : HISAMINATO YUJI
JORDI BONADA

(54) VOICE SYNTHESIZING DEVICE

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a voice synthesizing device which reduces the size of a database while minimizing deterioration in voice quality.

SOLUTION: The voice synthesizing device has a storage means which stores phoneme pieces having different pitches by phonemes represented as the same phoneme symbols, a readout means which reads the phoneme pieces out by using the pitches as indexes, and a voice synthesizing means which synthesizes a voice according to the read phoneme pieces.



LEGAL STATUS

[Date of request for examination]

19.06.2003

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2002-268659
(P2002-268659A)

(43) 公開日 平成14年9月20日 (2002.9.20)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード [*] (参考)
G 1 0 L 13/00		G 0 6 F 3/16	3 3 0 K 5 D 0 4 5
G 0 6 F 3/16	3 3 0	G 1 0 L 3/00	J
G 1 0 L 13/08			H
13/06		5/04	F
		9/02	L
審査請求 未請求 請求項の数13 O L (全 20 頁)			

(21) 出願番号 特願2001-67258(P2001-67258)

(22) 出願日 平成13年3月9日(2001.3.9)

(71) 出願人 000004075

ヤマハ株式会社

静岡県浜松市中沢町10番1号

(72) 発明者 久湊 裕司

静岡県浜松市中沢町10番1号 ヤマハ株式会社内

(72) 発明者 ジョルディ ボナダ

パッセイグデエ・シルクインバラシ
オ・8. 08003 パルセロナ スペイン

(74) 代理人 100091340

弁理士 高橋 敬四郎 (外2名)

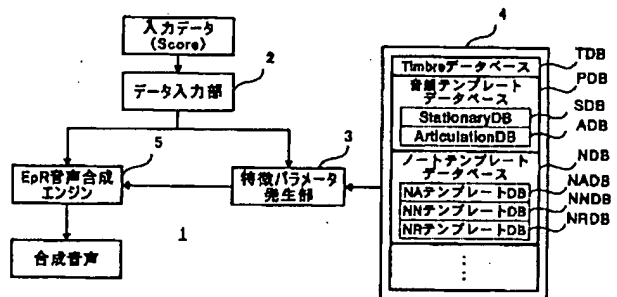
Fターム(参考) 5D045 AA07 AA20

(54) 【発明の名称】 音声合成装置

(57) 【要約】

【課題】 音質の劣化を最小限に抑えつつ、データベースのサイズを縮小した音声合成装置を提供する。

【解決手段】 音声合成装置は、同じ音素記号で表される音素毎に複数の異なるピッチを持つ音素片を記憶する記憶手段と、前記音素片をピッチをインデックスとして読み出す読み出し手段と、前記読み出された音素片に基づき音声を合成する音声合成手段とを有する。



【特許請求の範囲】

【請求項1】 同じ音素記号で表される音素毎に複数の異なるピッチを持つ音素片を記憶する記憶手段と、前記音素片をピッチをインデックスとして読み出す読出し手段と、

前記読み出された音素片に基き音声を作成する音声合成手段とを有する音声合成装置。

【請求項2】 同じ音素記号で表される音素毎に複数の異なる音楽表現度を持つ音素片を記憶する記憶手段と、前記音素片を音楽表現度をインデックスとして読み出す読出し手段と、

前記読み出された音素片に基き音声を作成する音声合成手段とを有する音声合成装置。

【請求項3】 同じ音素記号で表される音素毎に複数の異なる音素片を記憶する記憶手段と、

音声合成の為の音声情報を入力する入力手段と、前記音声情報に合致する音素片が前記記憶手段に記憶されていない場合に、前記記憶手段に記憶されている音素片を元に前記音声情報に合致する音素片を補間して算出する補間手段と、

前記補間して算出された音素片に基き音声を作成する音声合成手段とを有する音声合成装置。

【請求項4】 音声の特徴量の時間変化分をテンプレートデータとして記憶する記憶手段と、

音声合成の為の音声情報を入力する入力手段と、前記音声情報に基き前記テンプレートデータを前記記憶手段から読み出す読出し手段と、

前記読み出されたテンプレートデータと、前記音声情報に基き音声を作成する音声合成手段とを有する音声合成装置。

【請求項5】 同じ音素記号で表される音素毎に複数の異なるピッチを持つ音素片を記憶する記憶手段から、ピッチをインデックスとして音素片を読み出す読出し工程と前記読み出された音素片に基き音声を作成する音声合成工程とを有する音声合成方法。

【請求項6】 同じ音素記号で表される音素毎に複数の異なる音楽表現度を持つ音素片を記憶する記憶手段から、音楽表現度をインデックスとして音素片を読み出す読出し工程と前記読み出された音素片に基き音声を作成する音声合成工程とを有する音声合成方法。

【請求項7】 同じ音素記号で表される音素毎に複数の異なる音素片を記憶する記憶手段から音素片を読み出す読出し工程と、

音声合成の為の音声情報を入力する入力工程と、前記音声情報に合致する音素片が前記記憶手段に記憶されていない場合に、前記記憶手段に記憶されている音素片を元に前記音声情報に合致する音素片を補間して算出する補間工程と、

前記補間して算出された音素片に基き音声を作成する音声合成工程とを有する音声合成方法。

【請求項8】 音声の特徴量の時間変化分をテンプレートデータとして記憶する記憶工程と、

音声合成の為の音声情報を入力する入力工程と、前記音声情報に基き音声の特徴量の時間変化分を表すテンプレートデータを前記記憶手段から読み出す読出し工程と、

前記読み出されたテンプレートデータと、前記音声情報に基き音声を作成する音声合成工程とを有する音声合成方法。

【請求項9】 同じ音素記号で表される音素毎に複数の異なるピッチを持つ音素片を記憶する記憶手段から、ピッチをインデックスとして音素片を読み出す読出し手順と前記読み出された音素片に基き音声を作成する音声合成手順とを有する音声合成手順をコンピュータに実行させるためのプログラム。

【請求項10】 同じ音素記号で表される音素毎に複数の異なる音楽表現度を持つ音素片を記憶する記憶手段から、音楽表現度をインデックスとして音素片を読み出す読出し手順と前記読み出された音素片に基き音声を作成する音声合成手順とを有する音声合成手順をコンピュータに実行させるためのプログラム。

【請求項11】 同じ音素記号で表される音素毎に複数の異なる音素片を記憶する記憶手段から音素片を読み出す読出し手順と、

音声合成の為の音声情報を入力する入力手順と、前記音声情報に合致する音素片が前記記憶手段に記憶されていない場合に、前記記憶手段に記憶されている音素片を元に前記音声情報に合致する音素片を補間して算出する補間手順と、

前記補間して算出された音素片に基き音声を作成する音声合成手順とを有する音声合成手順をコンピュータに実行させるためのプログラム。

【請求項12】 音声の特徴量の時間変化分をテンプレートデータとして記憶する記憶手段と、

音声合成の為の音声情報を入力する入力手段と、前記音声情報に基き音声の特徴量の時間変化分を表すテンプレートデータを記憶手段から読み出す読出し手順と、

前記読み出されたテンプレートデータと、前記音声情報に基き音声を作成する音声合成手段とを有する音声合成手順をコンピュータに実行させるためのプログラム。

【請求項13】 同じ音素記号で表される音素毎に複数の異なるピッチを持つ音素片と、

音声の特徴量の時間変化分を表すテンプレートデータであって、前記音素片に対して適用されるテンプレートデータとを有する音声合成用データベースを記憶した媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、音声合成装置に関

し、より詳しくは、人間の歌唱音声を作成する音声合成装置に関する。

【0002】

【従来の技術】人間の音声は、音韻（音素）により構成され、各音韻は複数のフォルマントにより構成されている。よって、人間の歌唱音声の合成は、まず、人間が発生することのできる全ての音韻に対して、その各音韻を構成する全てのフォルマントを発生して合成できるように準備し、必要な音韻を生成する。次に、生成された複数の音韻を順次つなぎ合わせ、メロディに合わせて音高を制御する。この手法は、人間の音声に限らず、フォルマントを有する楽音、例えば、管楽器から発生される楽音の合成にも適用できる。

【0003】この手法を用いた音声合成装置は従来から知られており、例えば、特許公報第2504172号には、高い音高のフォルマント音を発生するときでも、不要なスペクトルを発生しないように構成したフォルマント音発生装置が開示されている。

【0004】また、フォルマント周波数は、ピッチに依存することが知られており、特開平6-308997号公報の実施例に記載されているように、ピッチ周波数ごとにいくつかの音素片をデータベースに持っておき、音声のピッチに従って、適切な音素片を選択する技術が知られている。

【0005】

【発明が解決しようとする課題】しかし、上記のような従来のデータベースでは、1つの音素片について、一定以上数のピッチ周波数の音素片を持つ必要があり、データベースのサイズが、比較的大きくなってしまふ。

【0006】また、多くの異なるピッチで発生された音声から音素片を抽出する必要があるために、データベースの構築に時間を要する。

【0007】さらには、フォルマント周波数は、ピッチのみに依存するのではなく、他の要素、例えば、ダイナミクス等が加わることに伴い、二乗、三乗とデータ量が増えてしまふ。

【0008】本発明の目的は、音質の劣化を最小限に抑えつつ、データベースのサイズを縮小した音声合成装置を提供することである。

【0009】また、本発明の他の目的は、上記データベースを用いた音声合成装置を提供することである。

【0010】

【課題を解決するための手段】本発明の一観点によれば、音声合成装置は、同じ音素記号で表される音素毎に複数の異なるピッチを持つ音素片を記憶する記憶手段と、前記音素片をピッチをインデックスとして読み出す読出し手段と、前記読み出された音素片に基づき音声を合成する音声合成手段とを有する。

【0011】また、本発明の他の観点によれば、音声合成装置は、同じ音素記号で表される音素毎に複数の異なる

音楽表現度を持つ音素片を記憶する記憶手段と、前記音素片を音楽表現度をインデックスとして読み出す読出し手段と、前記読み出された音素片に基づき音声を合成する音声合成手段とを有する。

【0012】また、本発明の他の観点によれば、音声合成装置は、同じ音素記号で表される音素毎に複数の異なる音素片を記憶する記憶手段と、音声合成の為の音声情報を入力する入力手段と、前記音声情報に合致する音素片が前記記憶手段に記憶されていない場合に、前記記憶手段に記憶されている音素片を元に前記音声情報に合致する音素片を補間して算出する補間手段と、前記補間して算出された音素片に基づき音声を合成する音声合成手段とを有する。

【0013】また、本発明の他の観点によれば、音声合成装置は、音声の特徴量の時間変化分をテンプレートデータとして記憶する記憶手段と、音声合成の為の音声情報を入力する入力手段と、前記音声情報に基づき前記テンプレートデータを前記記憶手段から読み出す読出し手段と、前記読み出されたテンプレートデータと、前記音声情報に基づき音声を合成する音声合成手段とを有する。

【0014】

【発明の実施の形態】図1は、音声合成装置1の構成を表すブロック図である。

【0015】音声合成装置1は、データ入力部2、特徴パラメータ発生部3、データベース4、E p R音声合成エンジン5を有する。

【0016】データ入力部2に入力される入力データScoreは、特徴パラメータ発生部3及びE p R音声合成エンジン5に送られる。特徴パラメータ発生部3は、入力データScoreに基づきデータベース4から後述する特徴パラメータ、各種テンプレートを読み込む。特徴パラメータ発生部3は、さらに、読み込んだ特徴パラメータに各種テンプレートを適用して、最終的な特徴パラメータを生成してE p R音声合成エンジン5に送る。

【0017】E p R音声合成エンジン5では、入力データScoreのピッチ、ダイナミクス等に基づきパルスが発生させ、該発生させたパルスに特徴パラメータを適用することにより、音声を合成して出力する。

【0018】図2は、入力データScoreの一例を示す概念図である。音韻トラックPHT、ノートトラックNT、ピッチトラックPIT、ダイナミクストラックDYT、オープニングトラックOTによって構成されており、楽曲のフレーズ若しくは曲全体の、時間とともに変化するデータが保存されている楽曲データである。

【0019】音韻トラックPHTには、音韻名と、その発音継続時間が含まれる。さらに、各音韻は、音素と音素の遷移部分であることを示すアーティキュレーション（Articulation）とその他の定常部分であることを示すステーションナリー（Stationary）

y) との2つに分類される。各音韻は、これらのうちどちらに分類されるかに付いてのフラグも含むものとする。なお、アーティキュレーションは、遷移部分であるので、先頭音韻名と後続音韻名の複数の音韻名を有している。一方、ステーションナリーは定常部分であるので1つの音韻名だけからなる。

【0020】ノートトラックNTには、ノートアタック(Not eAtt ack)、ノートトゥノート(Not eToNot e)、ノートリリース(Not eRel e)のいずれかを示すフラグが記録されている。ノートアタックは発音の立ち上がり時、ノートトゥノートは音程の変化時、ノートリリースは発音の立下り時の音楽表現を指示するコマンドである。

【0021】ピッチトラックPITには、各時刻において発音すべき音声の基本周波数が記録されている。なお、実際に発音される音声のピッチはこのピッチトラックPITに記録されているピッチ情報に基づき他の情報を用いて算出されるので、実際に発音されているピッチと、ここに記録されているピッチは異なる場合がある。

【0022】ダイナミクストラックDYTには、音声の強さを示すパラメータである各時刻におけるダイナミクス値が記録されている。ダイナミクス値は、0から1までの値をとる。

【0023】オープニングトラックOTには、唇の開き具合(唇開度)を示すパラメータである各時刻のオープニング値が記録されている。オープニング値は0から1

$$ExcitationCurve(f) = EGain + ESlopeDepth * (exp(-ESlope * f) - 1)$$

… (A)

励起レゾナンスは、胸部による共鳴を表す。中心周波数(ERFreq)、バンド幅(ERBW)、アンプリチュード(ERamp)の3つのパラメータで構成され、2次フィルター特性を有している。

【0028】フォルマントは、1から12個のレゾナンスを組み合わせることで声道による共鳴を表す。中心周波数(FormantFreq)、バンド幅(FormantBW_i)、アンプリチュード(FormantAmp_i)の3つのパラメータで構成される。なお、「i」は、1から12までの値(1 ≤ i ≤ 12)である。

【0029】差分スペクトルは、上記の励起波形スペクトルのエンベロープ、励起レゾナンス、フォルマントの3つで表現することの出来ないオリジナルスペクトルとの差分のスペクトルを持つ特徴パラメータである。

【0030】データベース4は、少なくともTimbreデータベースTDB、音韻テンプレートデータベースPDB、ノートテンプレートデータベースNDBから構成されている。

【0031】一般に、TimbreデータベースTDBに保存されている特定の時刻から得られた特徴パラメータのみを用いて音声を合成した場合には非常に単調で、

までの値をとる。

【0024】特徴パラメータ発生部3は、データ入力部2から入力される入力データScoreに基づき、データベース4からデータを読み出し、後述するように、入力データScore及びデータベース4から読み出したデータに基づき特徴パラメータを発生して、EPR音声合成エンジン5に出力する。

【0025】この特徴パラメータ発生部3で発生する特徴パラメータは、例えば、励起波形スペクトルのエンベロープ、励起レゾナンス、フォルマント、差分スペクトルの4つに分類することが出来る。これらの4つの特徴パラメータは、実際の人間の音声等(オリジナルの音声)を分析して得られる調和成分のスペクトル・エンベロープ(オリジナルのスペクトル)を分解することにより得られるものである。

【0026】励起波形スペクトルのエンベロープ(ExcitationCurve)は、声帯波形の大きさ(dB)を表すEGain、声帯波形のスペクトルエンベロープの傾きを表すESlopeDepth、声帯波形のスペクトルエンベロープの最大値から最小値の深さ(dB)を表すESlopeの3つのパラメータによって構成されており、以下の式(A)で表すことが出来る。

【0027】

【数式1】

機械的な音声になる。また、音素が連続する場合にはその遷移部分での音声は実際には徐々に変化してゆくので、音素の定常部分のみを単純に連結した場合には、接続点では非常に不自然な音声となる。そこで音韻テンプレート、及びノートテンプレートをデータベースとして持ち、音声合成時に使用することにより、それらの欠点を低減することが可能となる。

【0032】Timberとは音韻の音色であり、ある時刻1点における特徴パラメータ(励起スペクトル、励起レゾナンス、フォルマント、差分スペクトルのセット)で表現される。図3にTimbreデータベースTDBの例を示す。このデータベースは、インデックスとして音韻名、ピッチを持つ。

【0033】なお、以下、この明細書では図3に示すTimbreデータベースTDBを使うが、より細かく特徴パラメータを指定できるように、図4に示すようにインデックスとして音韻名、ピッチ、ダイナミクス、オープニングの4つを持つデータベースを用意してもよい。

【0034】音韻テンプレートデータベースPDBはステーションナリーテンプレートデータベースとアーティキュレーションテンプレートデータベースで構成される。ここでテンプレートとは、特徴パラメータPとピッチPitchのペアが一定時間ごとに並んだシーケンス、及

び、その区間の長さ T (sec.) の組であり、以下の式 (B) で表すことができる。

【0035】

【数式2】

$$Template = \{P(t), Pitch(t), T\}$$

… (B)

なお、 $t=0, \Delta t, 2\Delta t, 3\Delta t, \dots, T$ であり、本実施例では、 Δt は5msとする。

【0036】 Δt を小さくすると時間分解能がよくなるので音質は良くなるがデータベースのサイズが大きくなり、逆に Δt を大きくすると音質が悪くなるがデータベースのサイズは小さくなる。 Δt を決定する際には音質とデータベースのサイズとの優先度を考慮して決定すればよい。

【0037】図5は、ステーションナリーテンプレートデータベースの一例である。ステーションナリーテンプレートデータベースは、音韻名と代表ピッチをインデックスとして、すべての有声音韻についてのステーションナリーテンプレートを有している。ステーションナリーテンプレートは音韻、ピッチの安定した部分の音声をEPRモデルを使って分析することによって得ることができる。

【0038】あるひとつの有声音、例えば「あ」、を長く伸ばして、ある音程、例えばC4、で発声した場合にはピッチやフォルマント周波数などの特徴パラメータは、ほぼ一定であり定常 (ステーションナリー) であると言えるが、実際には若干の変動が生じている。この変動がなく完全に一定の場合には無機質で機械的な音声になってしまい、逆に言えば、その変動が人間らしさ、自然性を表すと言える。

【0039】有声音を合成する場合に、Timbre、つまりある時刻1点の特徴パラメータのみを使うのではなく、それにステーションナリーテンプレートにある実際の人間の音声から取り出した特徴パラメータの時間変動分、ピッチ変動分を加算することによって有声音に自然性を与えることができる。

【0040】歌唱音声合成の場合には音符の長さに従って発音する時間を変化させる必要があるが、十分長いテンプレートを1つだけ用意する。テンプレートよりも長い有声音を合成する場合には、テンプレートの時間軸の伸縮をすることはしないで、テンプレートの持っている時間をそのままにして有声音の先頭部分からテンプレートを適用する。

【0041】テンプレートの終端まで達したら、その後再び同じテンプレートを繰り返し適用する。なお、テ

$$Template3 = \left\{ \begin{array}{l} P(t) - ((P(T) - P(0)) * t / T + P(0)), \\ Pitch(t) - ((Pitch(T) - Pitch(0)) * t / T + Pitch(0)), T \end{array} \right\}$$

… (C3)

人間が2つの音素を連続して発音する場合には、突然変

ンプレートの終端まで達したら、テンプレートの時間を逆にしたテンプレートを適用する方法も考えられる。この方法ではテンプレートの接続点での不連続がなくなる。

【0042】テンプレートの時間軸を伸縮することをしないのは、特徴パラメータ、ピッチの変動のスピードが大きく変わると自然性が損なわれるからである。定常部分の揺らぎは人間が意識してコントロールするものではないという考え方からも伸縮しない方が好ましい。

【0043】ステーションナリーテンプレートは、定常部分の特徴パラメータの時系列をそのまま持つのではなく、その音素の代表的な特徴パラメータと、その変動量を持つ構造である。定常部分の特徴パラメータの変動量は小さいことから、特徴パラメータをそのまま持つことに比べて、変動量で持つ方が情報量が少なく、データベースのサイズを小さくする効果がある。

【0044】図6はアーティキュレーションテンプレートデータベースの一例である。アーティキュレーションテンプレートデータベースは、先頭音韻名と後続音韻名と代表ピッチとをインデックスとしている。アーティキュレーションテンプレートデータベースには、一定の言語における現実的に可能な音韻の組合せについてアーティキュレーションテンプレートが保存されている。

【0045】アーティキュレーションテンプレートはピッチの安定した、音韻の接続部分の音声をEPRモデルを使って分析することによって得ることができる。

【0046】なお、特徴パラメータ $P(t)$ は絶対値そのままでいいが、差分値を用いることも出来る。後述するように、合成時には、これらのテンプレートの値の絶対値がそのまま利用されるのではなく、パラメータの相対的な変化量が利用されるので、テンプレートの適用方法に従って、以下の式 (C1) ~ (C3) に示すように $P(t=T)$ からの差分、あるいは $P(0)$ からの差分、あるいは $P(0)$ と $P(T)$ を直線で結んだ値との差分の形で特徴パラメータを記録する。

【0047】

【数式3】

$$Template1 = \{P(t) - P(T), Pitch(t) - Pitch(T), T\}$$

… (C1)

【数式4】

$$Template2 = \{P(t) - P(0), Pitch(t) - Pitch(0), T\}$$

… (C2)

【数式5】

化するのではなくゆるやかに移行していくので、例えば、「あ」という母音の後に区切りを置かないで連続し

て「え」という母音を発音する場合には、最初に「あ」が発音され「あ」と「え」の中間に位置する発音を経て「え」に変化する。

【0048】この現象は一般に調音結合と呼ばれる現象である。音素の結合部分が自然になるように音声合成を行うには、ある言語において組合せ可能な音素の組合せについて、結合部分の音声情報を何らかの形で持つことが好ましい。

【0049】音素の結合部分をLPC係数や音声波形といった形でそのまま持つ方式はすでに存在しているが、本実施例では、特徴パラメータ、ピッチの差分情報を持ったアーティキュレーションテンプレートをを使って2つの音素間の調音(Articulation)部分を合成している。

【0050】例えば、2つの連続する同じ音程の4分音符で、それぞれの歌詞が「あ」、「い」という歌唱を合成する場合を考える。2つの音符の境界には「あ」から「い」への移行部分が存在する。「あ」、「い」は両方とも母音であり、有声音であるので、V(有声音)からV(有声音)へのアーティキュレーションに該当し、後述するタイプ3の方法でアーティキュレーションテンプレートを適用して移行部分の特徴パラメータを求めることができる。

【0051】すなわち、「あ」と「い」の特徴パラメータをTimbreデータベースTDBから読み出し、それらに「あ」から「い」へのアーティキュレーションテンプレートを適用すれば、その移行部分の、自然な変化を持つ特徴パラメータが得られる。

【0052】ここで、「あ」から「い」への移行部分の時間を、その部分に適用するアーティキュレーションテンプレートの元々の時間と同じにすれば、テンプレートを作成するときに利用した音声波形と同じ変化を得る事が出来る。

【0053】テンプレートの時間よりもゆっくりと、あるいは長く変化する音声を合成する場合には、テンプレートの長さを線形に伸長してから特徴パラメータの差分を加算すればよい。先に説明したステーションナリと異なり、2つの音素間の変化部分のスピードは意識的にコントロールできるものであるため、線形にテンプレートを伸縮しても大きな不自然性は生じない。

【0054】次に2つの連続する同じ音程の4分音符で、それぞれの歌詞が「あ」、「す」という歌唱を合成する場合を考える。2つの音符の境界には「あ」から「す」の子音部分への短い移行部分が存在する。これはV(有声音)からU(無声音)へのアーティキュレーションに該当するので、後述するタイプ1の方法でアーティキュレーションテンプレートを適用することで移行部分の特徴パラメータを求めることができる。

【0055】「あ」の特徴パラメータをTimbreデータベースTDBより求めて、それに「a」から「s」

へのアーティキュレーションテンプレートを適用することで、自然な変化を持つ移行部分の特徴パラメータを得る事が出来る。

【0056】V(有声音)からU(無声音)へのアーティキュレーションで、タイプ1、つまりテンプレートの先頭部分からの差分、を使う理由は、単純に終端部分にあたるU(無声音)部分にはピッチ、特徴パラメータが存在しないためである。

【0057】「す」はローマ字であらわすと「su」であり、子音部分「s」と母音部分「u」から構成される。この中間点にも、「s」の音を残しながら「u」が発音される移行部分が存在する。これはUからVへのアーティキュレーションに該当するので、ここでもまたタイプ1の方法でアーティキュレーションテンプレートを適用する。

【0058】「う(u)」の特徴パラメータをTimbreデータベースTDBから読み出し、それに「s」から「u」へのアーティキュレーションテンプレートを適用することで、「s」から「u」への変化部分の特徴パラメータを得ることができる。

【0059】特徴パラメータの差分情報を持ったアーティキュレーションテンプレートは、絶対値で特徴パラメータを記録したテンプレートに比べて、データサイズが少なくなるという利点を持っている。

【0060】ノートテンプレートデータベースNDBは、少なくとも、ノートアタックテンプレート(NAテンプレート)データベースNADB、ノートリリーステンプレート(NRテンプレート)データベースNRDB、ノートトゥノートテンプレート(NNテンプレート)データベースNNDBを含んでいる。

【0061】図7はNAテンプレートデータベースNADBの一例である。NAテンプレートには音声の立ち上がり部分の特徴パラメータ及びピッチの変化情報が含まれている。

【0062】NAテンプレートデータベースNADBには、音韻名と代表ピッチをインデックスとして、すべての有声音の音韻についてのNAテンプレートが保存されている。NAテンプレートは、実際に発音した音声の立ち上がり部分を分析することによって得られる。

【0063】NRテンプレートには音声の立下り部分の特徴パラメータ及びピッチの変化情報が含まれている。NRテンプレートデータベースNRDBはNAテンプレートデータベースNADBと同じ構造であり、音韻名と代表ピッチをインデックスとして、すべての有声音の音韻についてのNRテンプレートを持っている。

【0064】一定のピッチである音素、例えば「あ」を発声しようとしたときの立ち上がり部分(Attack)を分析すると振幅が徐々に大きくなり、一定のレベルになって安定していくことがわかる。振幅値だけではなく、フォルマント周波数、フォルマントバンド幅、ピ

ッチについても変化している。

【0065】人間の実際に発声した音声、例えば「あ」、の立ち上がり部分を解析して得たNAテンプレートを、定常部分の特徴パラメータに通用することで、その立ち上がり部分の人の音声の持つ自然な変化を与えることができる。

【0066】すべての音素ごとにNAテンプレートを用意すれば、どの音素についてもアタック部分の変化を与えることが可能になる。

【0067】歌唱では、音楽的に表情をつけるために立ち上がりを速くしたり、ゆっくりと歌う場合がある。NAテンプレートは、あるひとつの立ち上がりの時間を持っているが、もともとNAテンプレートの持っている速さよりも速く、若しくは遅くすることは、テンプレートの時間軸を線形に伸縮してから適用することで可能になる。

【0068】テンプレートを伸縮しても、数倍の範囲内ならば、アタックに不自然さは生じないことが実験によりわかっている。より広範囲のアタックの長さを指定して合成できるようにするには、数段階の長さのNAテンプレートを用意して、最も長さの近いテンプレートを選択して伸縮するなどの方法を使う。

【0069】発声の終了する部分、つまり立下り(Release)についても、立ち上がり(Attack)と同様に振幅、ピッチ、フォルマントが変化する。

【0070】立下り部分に人間の音声の持つ自然な変化を与えるのは、人間が実際に発声した音声の立ち下り部分を解析して得たNRテンプレートを、立下りの開始する前の音素の特徴パラメータに対して適用することで可能となる。

【0071】図8は、NNテンプレートデータベースNNDBの一例である。NNテンプレートはピッチが変化する部分の音声の特徴パラメータを持っている。NNテンプレートデータベースNNDBには、音韻名、テンプレートの始点時刻のピッチ、終了時刻のピッチをインデックスとして、すべての有聲の音韻についてのNNテンプレートが保存されている。

【0072】ピッチの異なる2つの音符を連続して間を置かず歌唱するとき、前の音符の音程から、後ろの音符のピッチに滑らかにピッチを変化させながら歌う歌唱方法がある。ピッチやアンプリチュードが変化するのには当然であるが、さらに、前後2つの音符の発音が同じ(例えば同じ「あ」)だとしても、フォルマント周波数などの音声の周波数特性が微妙に変化する。

【0073】実際にピッチを変化させて歌った音声の変化を始点から終点まで解析して求めたNNテンプレートを使うことによって、そのような音程の異なる音符の境界に、自然な音楽的表現を、与えることができる。

【0074】実際の音楽における旋律では、2オクターブ24音の音域としたとしても、ピッチ変化の組合せは

非常に多い。しかし、実際にはピッチの絶対値が異なってもピッチ差が近いテンプレートで代用することができるので全ての組合せについてNNテンプレートを用意する必要はない。

【0075】NNテンプレートの選択においては、後述するように、ピッチの絶対値に近いものよりも、ピッチの変化幅が近いテンプレートを優先的に選択する。選択されたNNテンプレートは、後述するタイプ3の方法で適用する。

【0076】このとき、ピッチの変化幅が近いNNテンプレートを優先的に選ぶのは、ピッチの大きく変動する部分から作成したNNテンプレートには大きな値が入っている可能性があり、それをピッチの変化幅が少ない部分に適用した場合には元のNNテンプレートの持っている変化の形状を保てなくなり、変化が不自然になる可能性があるからである。

【0077】なお、ある特定の音素、例えば「あ」のピッチの変化している音声から求めたNNテンプレートを、全ての音素のピッチ変化に代用して使うことも可能であるが、データサイズが大きくても問題がない環境であれば、音素ごとに何パターンかピッチを変化させてNNテンプレートを用意するほうが、より単調でない豊かな合成音声が可能となる。

【0078】次に、データベース4に記録されているテンプレートの適用方法を説明する。テンプレートの適用とは、入力データScore上のある区間に対して、テンプレートの時間長を伸縮して、基準点となる1つ又は複数の特徴パラメータにテンプレートの特徴パラメータとの差分を加算して、Scoreのある区間と同じ時間長を持つ特徴パラメータ、ピッチの列を得ることである。具体的にはタイプ1からタイプ4までの4種類のテンプレートの適用方法がある。以下の説明ではテンプレートを{P(t), Pitch(t), T}であらわす。

【0079】まずタイプ1によるテンプレートの適用を説明する。タイプ1は、始点指定タイプによるテンプレートの適用方法である。入力データScoreの長さTの区間Kに対するタイプ1によるテンプレートの適用は、下記式(D)に従って時刻tでの特徴パラメータ P'_t を求めることである。なお P_t は区間Kの時刻tの特徴パラメータである。

【0080】

【数式6】

$$P'_t = P_t + P(t \cdot T / T') - P(0)$$

…(D)

なお、時刻 $t=0$ にテンプレート及び区間Kの始点があるとする。この式(D)はテンプレートの始点からの変化分を時刻tの特徴パラメータに加算することを意味する。

【0081】タイプ1は、テンプレートを主にノートリ

リース部分の特徴パラメータに適用する場合に用いる。何故なら、ノートリリースの開始部分では、定常部分の音声が存在する為、ノートリリースの開始部分でパラメータの連続性、つまりは音声の連続性を保つ必要があり、ノートリリースの終端部は無音であるので、その必要がないからである。

【0082】次にタイプ2によるテンプレートの適用方法を説明する。タイプ2は、終点指定タイプによるテンプレートの適用方法である。入力データScoreの長さTの区間Kに対するタイプ2によるテンプレートの適用は、下記式(E)に従って時刻tでの特徴パラメータ P'_t を求めることである。なお P_t は区間Kの時刻tの特徴パラメータである。

【0083】

【数式7】

$$P'_t = P_t + P(t \cdot T / T') - P(T)$$

…(E)

なお、時刻 $t=0$ にテンプレート及び区間Kの始点があるとする。この式(E)はテンプレートの終点からの変

$$P'_t = P_0 + \frac{t}{T'}(P_t - P_0) + (P(t \cdot T / T') - \frac{t}{T'}(P(T) - P(0)))$$

…(F)

なお、時刻 $t=0$ にテンプレート及び区間Kの始点があるとする。この式(F)はテンプレートの始点と終点を結んだ直線との差を、区間Kの始点と終点を結んだ直線に加算することを意味する。

【0087】次にタイプ4によるテンプレートの適用方法を説明する。タイプ4は、ステーションナリタイプによるテンプレートの適用方法である。入力データScoreの長さTの区間Kに対するタイプ2によるテンプレートの適用は、下記式(G)に従って時刻tでの特徴パラメータ P'_t を求めることである。なお P_t は区間Kの時刻tの特徴パラメータである。

【0088】

【数式9】

$$P'_t = P_t + P(t \bmod T) - P(0)$$

…(G)

なお、時刻 $t=0$ にテンプレート及び区間Kの始点があるとする。この式(G)は区間Kに対してテンプレートの始点からの特徴パラメータの変化分を加算することをT毎に繰り返すことを意味する。

【0089】タイプ4は、主にステーションナリ部分に適用する場合に用いる。このタイプ4は、比較的長時間の音声の定常的部分に自然な揺らぎを与える効果をもっている。

【0090】図9は、特徴パラメータ発生処理を表すフローチャートである。この処理により、ある時刻tにおける特徴パラメータを発生させる。この特徴パラメータ発生処理を、ある一定時刻毎に時刻tを増加させながら、繰り返し行うことにより、フレーズ、曲といった単

化分を時刻tの特徴パラメータに加算することを意味する。

【0084】タイプ2は、テンプレートを主にノートアタック部分の特徴パラメータに適用する場合に用いる。何故なら、ノートアタックの後方部分では、定常部分の音声が存在する為、ノートアタックの後方部分でパラメータの連続性、つまりは音声の連続性を保つ必要があり、ノートアタックの開始部分は無音であるので、その必要がないからである。

【0085】次にタイプ3によるテンプレートの適用方法を説明する。タイプ3は、両点指定タイプによるテンプレートの適用方法である。入力データScoreの長さTの区間Kに対するタイプ3によるテンプレートの適用は、下記式(F)に従って時刻tでの特徴パラメータ P'_t を求めることである。なお P_t は区間Kの時刻tの特徴パラメータである。

【0086】

【数式8】

位の音声を合成することが出来る。

【0091】ステップSA1では、特徴パラメータ発生処理を開始して次のステップSA2に進む。

【0092】ステップSA2では、入力データScoreの時刻tにおける各トラックの値を取得する。具体的には、入力データScore中の時刻tにおける音韻名、アーティキュレーション又はステーションナリの区別、ノートアタック、ノートトゥノート又はノートリリースの区別、ピッチ、ダイナミクス値、及びオープニング値を取得する。その後次のステップSA3に進む。

【0093】ステップSA3では、ステップSA2で取得した入力データScoreの各トラックの値に基づき、必要なテンプレートを音韻テンプレートデータベースPDBとノートテンプレートデータベースNDBから読み込む。その後次のステップSA4に進む。

【0094】このステップSA3での音韻テンプレートの読み込みは、例えば、以下の手順で行われる。時刻tでの音韻がアーティキュレーションであると判断すると、アーティキュレーションテンプレートデータベースを検索して、先頭と後続の音韻名が一致して、かつピッチが一番近いテンプレートを読み込む。

【0095】一方、時刻tでの音韻がステーションナリであると判断すると、ステーションナリテンプレートデータベースを検索して、音韻名が一致して、かつピッチが一番近いステーションナリテンプレートを読み込む。

【0096】また、ノートテンプレートの読み込みは、以下のように行われる。例えば、時刻tのノートトラックがノートアタックであると判断した場合は、NAテンプレートデータベースNADBを検索して、音韻名が一

致して、かつピッチが一番近いテンプレートを読み込む。

【0097】また、例えば、時刻 t のノートトラックがノートリリースであると判断した場合は、NRテンプレートデータベースNRDBを検索して、音韻名が一致して、かつピッチが一番近いテンプレートを読み込む。

【0098】さらに、例えば、時刻 t のノートトラックがノートトゥノートであると判断した場合は、NNテン

$$d = 0.8 \cdot |\text{TempInterval} - \text{Interval}| + 0.2 \cdot |\text{TempAve} - \text{Ave}|$$

… (H)

ここで、

$\text{TempInterval} = |\text{テンプレートの始点ピッチ} - \text{テンプレートの終点ピッチ}|$

$\text{TempAve} = (\text{テンプレートの始点ピッチ} + \text{テンプレートの終点ピッチ}) / 2$

$\text{Interval} = |\text{ノートトラック上の始点ピッチ} - \text{ノートトラック上の終点ピッチ}|$

$\text{Ave} = (\text{ノートトラック上の始点ピッチ} + \text{ノートトラック上の終点ピッチ}) / 2$

上記式 (H) で求めた距離 d に基づき、テンプレートを読み込むことにより、ピッチの絶対値が近いものよりも、ピッチの変化幅が近いテンプレートを優先的に選択するようにしている。

【0100】ステップSA4では、ノートトラックの現在時刻 t と同じ属性を持つ領域の開始時刻及び終了時刻を求め、音韻トラックがステーションナリーである場合はノートアタック、ノートトゥノート又はノートリリースの区別にしたがって、開始時刻あるいは終了時刻又は双方の特徴パラメータを取得若しくは算出する。その後次のステップSA5に進む。

【0101】時刻 t のノートトラックがノートアタックである場合には、TimbreデータベースTDBを検索して、音韻名及びノートアタック終了時刻のピッチが一致する特徴パラメータを読み込む。

【0102】ピッチが一致する特徴パラメータがないときには、音韻名が一致し、かつノートアタック終了時刻のピッチをはさむ2つの特徴パラメータを取得して、これらを補間することによりノートアタック終了時刻の特徴パラメータを算出する。補間方法の詳細は後述する。

【0103】時刻 t のノートトラックがノートリリースである場合には、TimbreデータベースTDBを検索して、音韻名及びノートアタック開始時刻のピッチが一致する特徴パラメータを読み込む。

【0104】ピッチが一致する特徴パラメータがないときには、音韻名が一致し、かつノートリリース開始時刻のピッチをはさむ2つの特徴パラメータを取得して、これらを補間することによりノートリリース開始時刻の特徴パラメータを算出する。補間方法の詳細は後述する。

【0105】時刻 t のノートトラックがノートトゥノートである場合には、TimbreデータベースTDBを検索して、音韻名とノートトゥノート開始時刻のピッチ

プレートデータベースNNDBを検索して、音韻名が一致して、かつ始点ピッチと終了時刻ピッチを元に以下の式 (H) で求められる距離 d が一番近くなるテンプレートを読み込む。以下の式 (H) は、周波数の変化量と平均値を重み付けして加算した値を元に距離尺度としている。

【0099】

【数式10】

【数式11】

が一致する特徴パラメータ及び音韻名とノートトゥノート終了時刻が一致する特徴パラメータを読み込む。

【0106】ピッチが一致する特徴パラメータがないときには、音韻名が一致し、かつノートトゥノート開始（終了）時刻のピッチをはさむ2つの特徴パラメータを取得して、これらを補間することによりノートトゥノート開始（終了）時刻の特徴パラメータを算出する。補間方法の詳細は後述する。

【0107】なお、音韻トラックがアーティキュレーションである場合は開始時刻及び終了時刻の特徴パラメータを取得若しくは算出する。この場合は、TimbreデータベースTDBを検索して、音韻名とアーティキュレーション開始時刻のピッチが一致する特徴パラメータ及び音韻名とアーティキュレーション終了時刻が一致する特徴パラメータを読み込む。

【0108】ピッチが一致する特徴パラメータがないときには、音韻名が一致し、かつアーティキュレーション開始（終了）時刻のピッチをはさむ2つの特徴パラメータを取得して、これらを補間することによりアーティキュレーション開始（終了）時刻の特徴パラメータを算出する。

【0109】ステップSA5では、ステップSA4で求めた始点、終了時刻の特徴パラメータとピッチに対して、ステップSA3で読み込んだテンプレートを適用して、時刻 t におけるピッチとダイナミクスを求める。

【0110】時刻 t のノートトラックがノートアタックならば、ノートアタック部分に対してステップSA4で求めたノートアタック部分の終了時刻の特徴パラメータを使いタイプ2でNAテンプレートを適用する。テンプレートを適用した後の時刻 t におけるピッチとダイナミクス (EGain) を記憶する。

【0111】一方、時刻 t のノートトラックがノートリ

リースならば、ノートリリース部分に対してステップSA4で求めたノートリリース始点の特徴パラメータを使いタイプ1でNRテンプレートを適用する。テンプレートを適用した後の時刻tにおけるピッチとダイナミクス(EGain)を記憶する。

【0112】また、時刻tのノートトラックがノートトゥノートならば、ノートトゥノート部分に対してステップSA4で求めたノートトゥノートの始点及び終了時刻における特徴パラメータを使い、その区間に対してタイプ3でNNテンプレートを適用する。テンプレートを適用した後の時刻tにおけるピッチとダイナミクス(EGain)を記憶する。

【0113】さらに、時刻tのノートトラックが上記のいずれでもない場合には、入力データScoreのピッチとダイナミクス(EGain)を記憶する。

【0114】以上のいずれかの処理を行ったら、次のステップSA6に進む。

【0115】ステップSA6では、ステップSA2で求めた各トラックの値から、時刻tの音韻がアーティキュレーションであるか否かを判断する。アーティキュレーションである場合には、YESの矢印で示すステップSA9に進む。アーティキュレーションでない場合、すなわち時刻tの音韻がステーションナリーである場合には、NOの矢印で示すステップSA7に進む。

【0116】ステップSA7では、ステップSA2で求めた時刻tにおける音韻名と、ステップSA5で求めたピッチ、ダイナミクスをインデックスとして、TimbreデータベースTDBから特徴パラメータを読み込み補間する。読み込みと補間の方法は、ステップSA4で行ったものと同様である。その後、ステップSA8に進む。

【0117】ステップSA8では、ステップSA7で求めた時刻tにおける特徴パラメータ及びピッチに対して、ステップSA3で求めたステーションナリーテンプレートをタイプ4で適用する。

【0118】このステップSA8で、ステーションナリーテンプレートを適用することで、時刻tでの特徴パラメータ及びピッチが更新され、ステーションナリーテンプレートの持つ音声の揺らぎが加えられる。その後、ステップSA10に進む。

【0119】ステップSA9では、ステップSA4で求めたアーティキュレーション部分の開始時刻及び終了時刻の特徴パラメータに、ステップSA3で読み込んだアーティキュレーションテンプレートを適用して、時刻tでの特徴パラメータ及びピッチを求める。その後、ステップSA10に進む。

【0120】ただし、テンプレートの適用方法は有声音(V)から無声音(U)への変化の場合はタイプ1で行い、無声音(U)から有声音(V)への変化の場合はタイプ2で行い、有声音(V)から有声音(V)又は無声

音(U)から無声音(U)への変化の場合はタイプ3で行う。

【0121】上記のようにテンプレートの適用方法を変えるのは、有声音部分での連続性を保ちつつ、テンプレートに含まれている自然な音声の変化を再現する為である。

【0122】ステップSA10では、ステップSA8若しくはステップSA9で求められた特徴パラメータに対して、NAテンプレート、NRテンプレート、NNテンプレートのいずれかを適用する。ただし、ここでは、特徴パラメータのEGainに対しては、テンプレートを適用しない。その後次のステップSA11に進み、特徴パラメータ発生処理を終了する。

【0123】このステップSA10でのテンプレートの適用は、時刻tでのノートトラックがノートアタックである場合には、ステップSA3で求めた、NAテンプレートをタイプ2により適用して、特徴パラメータを更新する。

【0124】時刻tでのノートトラックがノートリリースである場合には、ステップSA3で求めた、NRテンプレートをタイプ1により適用して、特徴パラメータを更新する。

【0125】時刻tでのノートトラックがノートトゥノートである場合には、ステップSA3で求めた、NNテンプレートをタイプ3により適用して、特徴パラメータを更新する。

【0126】ただし上記いずれの場合にも、ここでは、特徴パラメータのEGainに対しては、テンプレートを適用しない。また、ピッチについても、このステップSA10の前のステップで求められたものをそのまま使用する。

【0127】以下に、図9のステップSA4で行う特徴パラメータの補間について説明する。特徴パラメータの補間には、2つの特徴パラメータの補間と、1つの特徴パラメータからの推定がある。

【0128】人間が音声を発声するときにピッチを変化させると声帯波形(肺からの空気と声帯の振動によって発生する音源波形)が変化することが知られており、またフォルマントもピッチによって変化することが知られている。ある特定のピッチで歌った音声から得られた特徴パラメータを他のピッチの音声を合成するときそのまま流用した場合には、ピッチを変えても同じような声の音色になってしまい不自然になってしまう。

【0129】それを避けるために人間の歌唱音域である2~3オクターブの音域中、対数軸で、ほぼ等間隔で3点程度のピッチを選び、特徴パラメータをTimbreデータベースTDBに保存しておく。TimbreデータベースTDB中にあるピッチ以外のピッチの音声を合成する場合には、2つの特徴パラメータの補間(直線補間)若しくは1つの特徴パラメータからの推定(外挿)

によって特徴パラメータが求められる。

【0130】この方法によって、ピッチが変化したときの音声の特徴パラメータの変化を擬似的に表現することができる。また、ピッチの異なる特徴パラメータを3点程度持つのは、同じ音素、同じピッチの発生でもそのときによって特徴パラメータには変動があり、3点程度から補間して求めた場合とさらに細かく分割して求めた場合との差は余り意味がないからである。

$$P = \frac{|f_2 - f|}{|f_1 - f| + |f_2 - f|} P_1 + \frac{|f_1 - f|}{|f_1 - f| + |f_2 - f|} P_2 = P_1 + (P_2 - P_1) \frac{f - f_1}{f_2 - f_1}$$

… (I)

上記式 (I) では、データベースのインデックスがピッチ1個だけの場合を考えたが、一般的にインデックスがN個ある場合でも、目標を囲む近傍のN+1個のデータをもとに、以下の式 (I') を用いて、目標のインデックス f の代理として使用する特徴パラメータを補間して求めることが出来る。なお、P_iは、近傍のi番目の特徴パラメータであり、f_iはそのインデックスである。

【0133】

【数式13】

$$P = \sum_{i=1}^{N+1} \frac{\left(\sum_{j=1}^{N+1} |f_j - f| \right) - f_i}{\sum_{j=1}^{N+1} |f_j - f|} P_i$$

… (I')

1つの特徴パラメータからの推定は、データベースに含まれるデータの音域を外れる音声の特徴パラメータを推定するときに用いる。

【0134】これは、データベースの音域よりもピッチの高い音声を合成する場合に、データベース中の最もピッチの高い特徴パラメータをそのまま利用すると、明らかに音質が劣化するからである。

【0135】また、データベースの音域よりもピッチの低い音声を合成する場合に、最もピッチの低い特徴パラメータを利用すると同様に音質が劣化するからである。そこで本実施例では実際の音声データの観察からの知見に基づいた規則を使って、以下のように特徴パラメータを変化させて劣化を防いでいる。

【0136】まず、データベースの音域よりも高いピッチ (目標ピッチ) の音声を合成する場合を説明する。

【0137】まず、目標ピッチ TargetPitch [cents] からデータベース中の最も高いピッチ HighestPitch [cents] を引いた値 PitchDiff [cents] を求める。

【0138】次に、データベースから最も高いピッチを持つ特徴パラメータを読み出して、その内の励起レゾナンス周波数 EpRFreq 及び第 i フォルマント周波数

【0131】2つの特徴パラメータの補間は、例えば、2つの特徴パラメータとそれぞれのピッチの組 {P₁, f₁ [cents]}, {P₂, f₂ [cents]} が与えられたときに、時刻 t のピッチ f₁ [cents] における特徴パラメータを、以下の式 (I) により直線補間して求めることにより行われる。

【0132】

【数式12】

FormantFreq_iに、それぞれ上記 PitchDiff [cents] を加算して、EpRFreq', FormantFreq_i' に置き換えたものを目標ピッチの特徴パラメータとして使う。

【0139】次に、データベースの音域よりも低いピッチ (目標ピッチ) の音声を合成する場合を説明する。

【0140】まず、目標ピッチ TargetPitch [cents] からデータベース中の最も低いピッチ LowestPitch [cents] を引いた値 PitchDiff [cents] を求める。

【0141】次に、データベースから最も低いピッチを持つ特徴パラメータを読み出して、以下のようにパラメータを置き換えて目標ピッチの特徴パラメータとして用いる。

【0142】まず、励起レゾナンス周波数 EpRFreq 及び第1から第4フォルマント周波数 FormantFreq (1 ≤ i ≤ 4) を、それぞれ下記式 (J1) 及び (J2) を用いて、EpRFreq', FormantFreq_i' に置き換える。

【数式14】

$$ERFreq' = ERFreq + 0.25 \times PitchDiff$$

【数式15】

$$FormantFreq_i' = FormantFreq_i + 0.25 \times PitchDiff$$

さらに、ピッチが低くなるほどバンド幅が狭くなるように、励起レゾナンスバンド幅 ERBW 及び第1から第3フォルマントのバンド幅 FormantBW_i (1 ≤ i ≤ 3) をそれぞれ下記式 (J3)、(J4) の ERBW', FormantBW_i' に置き換える。

【0143】

【数式16】

$$ERBW' = \frac{ERBW}{1 - 3 \times PitchDiff / 1200}$$

… (J3)

【数式17】

$$FormantFreq_i' = FormantFreq_i + 0.25 \times PitchDiff$$

… (J4)

さらに、第1から第4フォルマントのアンブリチュード

FormantAmp1~FormantAmp4を下
記式(J5)~(J8)に従いPitchDiffに比
例させて大きくして、FormantAmp1'~Fo

rmantAmp4'に置き換える。

[0144]

【数式18】

$$FormantAmp_1' = FormantAmp_1 - 8 \times PitchDiff / 1200$$

…(J5)

【数式19】

$$FormantAmp_2' = FormantAmp_2 - 5 \times PitchDiff / 1200$$

…(J6)

【数式20】

$$FormantAmp_3' = FormantAmp_3 - 12 \times PitchDiff / 1200$$

…(J7)

【数式21】

$$FormantAmp_4' = FormantAmp_4 - 15 \times PitchDiff / 1200$$

…(J8)

さらに、スペクトル・エンベロープの傾きESlope
を下記式(J9)に従いESlope'に置き換える。

[0145]

【数式22】

$$ESlope' = ESlope \times (1 - 4 \times PitchDiff / 1200)$$

…(J9)

図4に示すような、ピッチ、ダイナミクス、オープニン
グをインデックスとしてTimbreデータベースTDB
を作成することが好ましいが、時間的、データベース
サイズの制約がある場合には、本実施例のように、図
3に示すような、ピッチのみをインデックスとしたデー
タベースを用いることになる。

【0146】そのような場合に、ダイナミクス関数や、
オープニング関数を用いて、ピッチのみをインデックス
とした特徴パラメータを変化させ、あたかも、ピッチ、
ダイナミクス、オープニングをインデックスとして作成
したTimbreデータベースTDBを使用したかのような効果
を擬似的に得る事が出来る。

【0147】すなわち、ピッチのみを変化させて録音し
た音声を使用して、ピッチ、ダイナミクス、オープニン
グを変化させて録音した音声を使用したかのような効果
を得る事が出来る。

【0148】ダイナミクス関数及び、オープニング関数
は、ダイナミクス、オープニングを変化させて発声した
実際の音声と、特徴パラメータの相関関係を分析して得
る事が出来る。以下に、ダイナミクス関数及び、オープ
ニング関数の例をあげ、その適用方法を説明する。

【0149】図10は、ダイナミクス関数の一例を表す
グラフである。図10(A)は、関数fEGを表すグラフ
であり、図10(B)は、関数fESを表すグラフであ
り、図10(C)は、関数fESDを表すグラフであ
る。

【0150】これらの、図10(A)~(C)に示され
る関数fEG、fES、fESDを利用して、ダイナミ
クス値を特徴パラメータExcitationGain
(EG)、ExcitationSlope(ES)、
ExcitationSlopeDepth(ESD)
に反映させる。

【0151】図10(A)~(C)の関数fEG、fES
S、fESDの入力は、全てダイナミクス値であり、0
から1までの値をとる。このダイナミクス値をdynと
して、関数fEG、fES、fESDを使い、下記式
(K1)~(K3)で、特徴パラメータEG'、E
S'、ESD'を求め、ダイナミクス値(dyn)の時
の特徴パラメータとして用いる。

【0152】

【数式23】

$$EG' = fEG(dyn)$$

…(K1)

【数式24】

$$ES' = ES \times fES(dyn)$$

…(K2)

【数式25】

$$ESD' = ESD + fESD(dyn)$$

…(K3)

なお、図10(A)~(C)の関数fEG、fES、f
ESDは、一例であり、歌唱者によって様々な関数を用
意することにより、より自然性を持った音声合成を行う
ことが出来る。

【0153】図11は、オープニング関数の一例を表す
グラフである。図中、横軸は周波数(Hz)であり、縦
軸はアンプリチュード(dB)である。

【0154】このオープニング関数をfOpen(fr
eq)とし、オープニング値をOpenとして、以下の
式(L1)により、励起レゾナンス周波数ERFreq
q'を励起レゾナンス周波数ERFreqから求め、オ
ープニング値(Open)のときの特徴パラメータとし
て用いる。

【0155】

【数式26】

$$ERFreq' = ERFreq + fOpen(ERFreq) \times (1 - Open)$$

…(L1)

また、以下の式(L2)により、i番目のフォルマント
周波数FormantFreqi'をi番目のフォルマ
ント周波数FormantFreqiから求め、オ
ープニング値(Open)のときの特徴パラメータとして用
いる。

【0156】

【数式27】

$$FormantFreq_i' = FormantFreq_i + fOpen(FormantFreq_i) \times (1 - Open)$$

… (L2)

これにより、周波数0～500Hzにあるフォルマントのアンプリチュードをオープニング値に比例させて増減させることができ、合成音声に、唇開度による音声の変化を与えることが出来る。

【0157】なお、オープニング値を入力とする関数を歌唱者別に用意して、変化させることにより、合成音声により多様化させることが出来る。

【0158】図12は、本実施例によるテンプレートの第1の適用例を表す図である。図中(a)の楽譜による歌唱を本実施例により合成する場合を説明する。

【0159】この楽譜は、最初の2分音符の音程は「ソ」であり、強さは「ピアノ(弱く)」で「あ」という発音である。2つ目の2分音符の音程は「ド」であり、強さは「メゾフォルテ(やや強く)」で「あ」という発音である。2つの2分音符は、レガートで接続されているので、音と音の間に切れ目がなく滑らかに接続する。

【0160】ここで、「ソ」から「ド」への変化の時間は、入力データ(楽譜)とともに与えられるものとする。

【0161】まず、音符の音名から2つのピッチの周波数が得られる。その後、2つのピッチの終点と始点を直線で結んで、図中(b)に示すように音符の境界部分のピッチを得ることが出来る。

【0162】次にダイナミクスであるが、これは、「ピアノ(弱く)」や「メゾフォルテ(やや強く)」といった強弱記号に対応した値をテーブルとして記憶しておき、これを使って数値に変換して2つの音符に対応するダイナミクス値を得る。このようにして得た2つのダイナミクス値を直線で結ぶことにより、図中(b)に示すように音符の境界部分のダイナミクス値を得ることが出来る。

【0163】このようにして得て、ピッチと、ダイナミクス値をそのまま用いると、ピッチ、ダイナミクスが音符の境界部分で急激に変化してしまうので、レガートに接続する為、この音符の境界部分に、図中(b)に示すようにNNテンプレートを適用する。

【0164】ここでは、ピッチとダイナミクスにだけ、NNテンプレートを適用して、図中(c)に示すような音符の境界部分が滑らかに接続されたピッチとダイナミクスを得る。

【0165】次に、図中(c)に示す決定されたピッチとダイナミクス及び「あ」という音韻名をインデックスとして、TimbreデータベースTDBから、図中(d)に示すような各時刻の特徴パラメータを求める。

【0166】ここで求めた各時刻の特徴パラメータに対

して、図中(c)に示す音韻名「あ」に対応するステーションナリーテンプレートを適用し、音符境界の接続部分以外の定常部分に音声の揺らぎを付加して、図中(e)に示すような特徴パラメータを得る。

【0167】次に、図中(b)でピッチとダイナミクスのみ適用したNNテンプレートの残り(フォルマント周波数など)を、図中(e)に示す特徴パラメータに適用し、音符の境界部分のフォルマント周波数などに揺らぎを与えた図中(f)で示す特徴パラメータを得る。

【0168】最後に、図中(c)のピッチ、ダイナミクスと、図中(f)の特徴パラメータを用いて、音声合成を行うことにより、図中(a)の楽譜で表す歌唱を合成することが出来る。

【0169】なお、図12の(b)で、NNテンプレートを適用する部分の時間幅は、例えば、図13に示すように長くすることが出来る。図13に示すように、NNテンプレートを適用する部分の時間幅を長くすると、NNテンプレートが伸長されて適用されるので、ゆっくりとした変化を持つ歌唱音声を合成することが出来る。

【0170】また、逆に、NNテンプレートを適用する時間幅を狭くすれば、早く滑らかに変化する歌唱音声を合成することが出来る。このようにNNテンプレートの適用時間を制御することで、変化のスピードをコントロールすることが出来る。

【0171】また、同じ時間で、ピッチをある高さから別の高さに変化させる場合でも、前半で急激に変化させ、後半はゆっくり変化させる歌い方があり、その逆もある。このように、ピッチの変化の道筋は何通りもあり、その違いは結果的に音楽的な聞こえ方の違いとなって現れる。そこで、このようなレガートの歌い方を変えて歌唱した音声から複数種類のNNテンプレートを作成して記録しておけば、様々なバリエーションを合成音声に持たせることが出来る。

【0172】さらに、音程(ピッチ)の変化の仕方には、上記のレガート奏法以外にも様々なものがあり、それらについても別にテンプレートを作成して記録するようにしてもよい。

【0173】例えば、レガートのように完全に連続的にピッチを変化させるのではなく、半音ごとにピッチを変化させたり、楽曲の長で使われる音階(例えば、ハ長調では、ドレミファソラシド)だけで飛び飛びに変化させたりする、いわゆるグリッサンド奏法がある。

【0174】この場合には、グリッサンドで実際に歌唱した音声から、NNテンプレートを作成し、そのテンプレートを適用して2つの音符を滑らかに接続した歌唱を合成することが出来る。

【0175】なお、本実施例では、NNテンプレート

は、同じ音韻でピッチが変化している場合だけを作成して記録しているが、例えば、「あ」から「え」のように違う音韻でピッチが変化している場合についても作成することができる。この場合は、NNテンプレートの数が多くなってしまいが、実際の歌唱により近づけることが出来る。

【0176】図14は、本実施例によるテンプレートの第2の適用例を表す図である。図中(a)の楽譜による歌唱を本実施例により合成する場合を説明する。

【0177】この楽譜は、最初の2分音符の音程は「ソ」であり、強さは「ピアノ(弱く)」で「あ」という発音である。2つ目の2分音符の音程は「ド」であり、強さは「メゾフォルテ(やや強く)」で「え」という発音である。

【0178】ここで、「あ」から「え」へのアーティキュレーションの時間は、2つの音素の組合せ毎に固定値として設定しておくか、又は入力データとともに与えられるものとする。

【0179】まず、音符の音名から2つのピッチの周波数が得られる。その後、2つのピッチの終点と始点を直線で結んで、図中(b)に示すように音符の境界部分(アーティキュレーション部分)のピッチを得ることが出来る。

【0180】次にダイナミクスであるが、これは、「ピアノ(弱く)」や「メゾフォルテ(やや強く)」といった強弱記号に対応した値をテーブルとして記憶しておき、これを使って数値に変換して2つの音符に対応するダイナミクス値を得る。このようにして得た2つのダイナミクス値を直線で結ぶことにより、図中(b)に示すように音符の境界部分のダイナミクス値を得ることが出来る。

【0181】次に、図中(b)に示す決定されたピッチとダイナミクス及び「あ」、「え」という音韻名をインデックスとして、TimbreデータベースTDBから、図中(c)に示すような各時刻の特徴パラメータを求める。ただし、アーティキュレーション部分の特徴パラメータは、仮に音韻「あ」の終点部分と、音韻「え」の始点部分を直線補間した値である。

【0182】次に、図中(c)に示すように、「あ」のステーションナリーテンプレート、「あ」から「え」へのアーティキュレーションテンプレート、「え」のステーションナリーテンプレートを先に求めた、特徴パラメータのそれぞれの該当部分に適用し、図中(d)に示すような特徴パラメータを得る。

【0183】最後に、図中(b)のピッチ、ダイナミクスと、(d)の特徴パラメータを使って、音声合成を行う。

【0184】このようにすると、人間が実際に発声する場合と同様に、自然に「あ」から「え」に変化する歌唱音声を合成することが出来る。

【0185】なお、アーティキュレーションテンプレートも、NNテンプレートの場合と同様に、境界部分(アーティキュレーション部分)の長さを楽譜とともに与えられるようにしておけば、「あ」から「え」へのアーティキュレーションの時間を制御することができ、ゆっくりと変化する音声や、早く変化する音声を、1つのテンプレートを伸縮することで合成できる。すなわち、こうすることで、音韻の変化する時間を制御することが出来る。

【0186】図15は、本実施例によるテンプレートの第3の適用例を表す図である。図中(a)の楽譜による歌唱を本実施例により合成する場合を説明する。

【0187】この楽譜は、音程が「ソ」で、発音は「あ」である全音符の強さを立ち上がりから次第に強くしていき、立下りで次第に弱くしていくものである。

【0188】この楽譜の場合は、ピッチ、ダイナミクスは図中(b)に示すように平坦である。これらのピッチ、ダイナミクスの先頭にNAテンプレートを適用し、さらに音符の最後にNRテンプレートを適用して、図中(c)で示すようなピッチとダイナミクスを求めて、決定する。

【0189】なお、NAテンプレート及びNRテンプレートを適用する長さは、クレッシェンド記号及びデクレッシェンド記号自身に長さを持たせて入力されているものとする。

【0190】次に、決定した図中(c)のピッチ、ダイナミクス及び音韻名「あ」をインデックスとして、図中(d)に示すようにアタックでもリリースでもない通常部分の特徴パラメータが求められる。

【0191】さらに、図中(d)に示す通常部分の特徴パラメータに、ステーションナリーテンプレートを適用して、図中(e)に示すような、揺らぎが与えられた特徴パラメータを求める。この(e)の特徴パラメータを元に、アタック部分とリリース部分の特徴パラメータを求める。

【0192】アタック部分の特徴パラメータは、通常部分の始点(アタック部分の終点)に対して、音韻「あ」のNAテンプレートを前述のタイプ2の方法で適用して求める。

【0193】リリース部分の特徴パラメータは、通常部分の終点(リリース部分の始点)に対して、音韻「あ」のNRテンプレートを前述のタイプ1の方法で適用して求める。

【0194】このようにして、アタック部分、通常部分、リリース部分の特徴パラメータが、図中(f)のように求められる。この特徴パラメータと、(c)のピッチ、ダイナミクスを使用して、音声を合成することで、(a)の楽譜によるクレッシェンド、デクレッシェンドで歌った歌唱音声を得ることが出来る。

【0195】以上、本実施例に拠れば、実際の人間の歌

唱音声を分析して得られる音韻テンプレートを用いて、特徴パラメータに変動を与えるので、歌唱音声の持っている母音を長く伸ばした部分や、音韻が変化する部分の特徴を反映した自然な合成音声を生成することが出来る。

【0196】また、本実施例に拠れば、実際の人間の歌唱音声を分析して得られるノートテンプレートを用いて、特徴パラメータに変動を与えるので、単なる音量の違いだけでなく、音楽的な強弱の表現力を持った合成音声を生成することが出来る。

【0197】さらに、本実施例に拠れば、ピッチ、ダイナミクス、オープニングなどの音楽表現度を細かく変化したデータを用意しなくても、他に用意されているデータを補間して、用いることが出来るので、少ないサンプルですみ、データベースのサイズを小さくすることが出来るとともに、データベースの作成時間を短縮することが出来る。

【0198】さらに、また、本実施例に拠れば、音楽表現度として、ピッチのみをインデックスとしたデータベースを使用したとしても、オープニング及びダイナミクス関数を用いて、擬似的にピッチ、オープニング、ダイナミクスの3つの音楽表現度をインデックスとして持つデータベースを使用した場合に近い効果を得る事が出来る。

【0199】なお、本実施例では、図2に示したように、入力データScoreとして、音韻トラックPH、ノートトラックNT、ピッチトラックPIT、ダイナミクストラックDYT、オープニングトラックOTを入力したが、入力データScoreの構成はこれに限られない。

【0200】例えば、図2の入力データScoreに、ビブラートトラックを追加して用意してもよい。ビブラートトラックには、0～1のビブラート値が記録されている。

【0201】この場合、データベース4には、ビブラート値を引数として、ピッチ、ダイナミクスの時系列を返す関数、若しくはテーブルをビブラートテンプレートとして保存しておく。

【0202】そして、図4のステップSA5のピッチ、ダイナミクスの計算において、このビブラートテンプレートを適用することで、ビブラート効果を与えたピッチ、ダイナミクスを得る事が出来る。

【0203】ビブラートテンプレートは、実際の人間の歌唱音声を分析することで得る事が出来る。

【0204】なお、本実施例は歌唱音声合成を中心に説明したが、歌唱音声に限られるものではなく、通常の会話の音声や楽器音なども同様に合成することができる。

【0205】なお、本実施例は、本実施例に対応するコンピュータプログラム等をインストールした市販のコンピュータ等によって、実施させるようにしてもよい。

【0206】その場合には、本実施例に対応するコンピュータプログラム等を、CD-ROMやフロッピー（登録商標）ディスク等の、コンピュータが読み込むことが出来る記憶媒体に記憶させた状態で、ユーザに提供してもよい。

【0207】そのコンピュータ等が、LAN、インターネット、電話回線等の通信ネットワークに接続されている場合には、通信ネットワークを介して、コンピュータプログラムや各種データ等をコンピュータ等に提供してもよい。

【0208】以上実施例に沿って本発明を説明したが、本発明はこれらに制限されるものではない。例えば、種々の変更、改良、組合せ等が可能なことは当業者に自明であろう。

【0209】

【発明の効果】以上説明したように、本発明によれば、音質の劣化を最小限に抑えつつ、サイズを縮小した音声合成用データベースを提供することができる。

【0210】また、本発明によれば、よりリアルな人間の歌唱音声を合成して、違和感のない自然な状態で歌を歌わせることが可能な音声合成装置を提供することができる。

【図面の簡単な説明】

【図1】 本発明の実施例による音声合成装置1の構成を表すブロック図である。

【図2】 入力データScoreの一例を示す概念図である。

【図3】 TimbreデータベースTDBの一例である。

【図4】 TimbreデータベースTDBの他の例である。

【図5】 ステーションナリーテンプレートデータベースの一例である。

【図6】 アーティキュレーションテンプレートデータベースの一例である。

【図7】 NAテンプレートデータベースNADBの一例である。

【図8】 NNテンプレートデータベースNNDBの一例である。

【図9】 特徴パラメータ発生処理を表すフローチャートである。

【図10】 ダイナミクス関数の一例を表すグラフである。

【図11】 オープニング関数の一例を表すグラフである。

【図12】 本実施例によるテンプレートの第1の適用例を表す図である。

【図13】 本実施例によるテンプレートの第1の適用例の変形例を表す図である。

【図14】 本実施例によるテンプレートの第2の適用

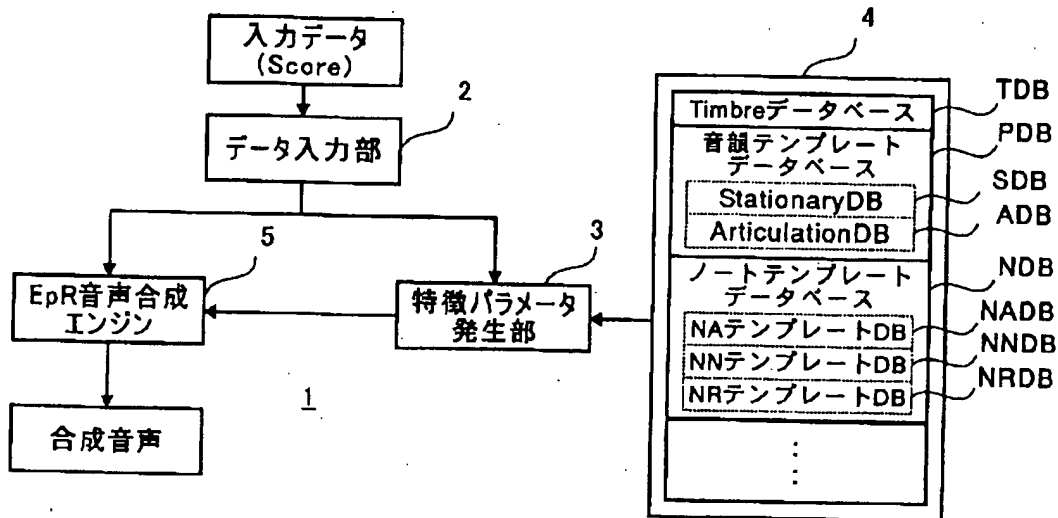
例を表す図である。

【図15】 本実施例によるテンプレートの第3の適用例を表す図である。

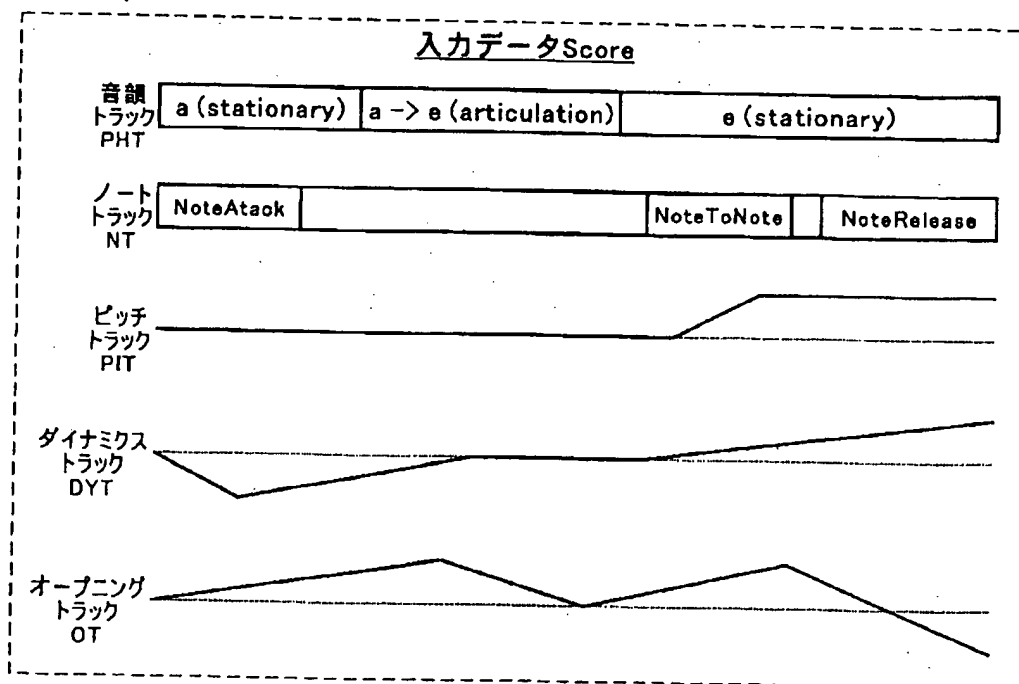
【符号の説明】

1…音声合成装置、2…データ入力部、3…特徴パラメータ発生部、4…データベース、5…EpR音声合成エンジン

【図1】



【図2】



【図3】

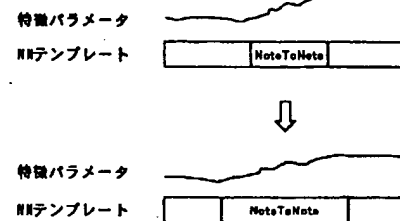
音韻名	ピッチ[Hz]	特徴パラメータ	音韻名	ピッチ[Hz]	ダイナミクス	オープニング	特徴パラメータ
/a/	200	Pa1	/a/	200	0.8	0.4	Pa1
/a/	300	Pa2	/a/	300	0.5	0.2	Pa2
/a/	400	Pa3	/a/	400	0.6	0.8	Pa3
/i/	200	Pi1	/i/	200	0.5	1	Pi1
/i/	300	Pi2	/i/	300	0.3	0.7	Pi2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
/o/	500	Po4	/o/	500	0.2	0.5	Po4

【図4】

【図5】

音韻名	代表ピッチ[Hz]	特徴パラメータ
/a/	200	[Pa1(t), Pitch_a1(t), Ta1]
/a/	300	[Pa2(t), Pitch_a2(t), Ta2]
/a/	400	[Pa3(t), Pitch_a3(t), Ta3]
/i/	200	[Pi1(t), Pitch_i1(t), Ti1]
/i/	300	[Pi2(t), Pitch_i2(t), Ti2]
⋮	⋮	⋮
/o/	500	[Po4(t), Pitch_o4(t), To4]

【図13】



【図6】

先頭音韻名	後続音韻名	代表ピッチ[Hz]	特徴パラメータ
/a/	/i/	200	[Pai1(t), Pitch_ai1(t), Tai1]
/a/	/i/	400	[Pai2(t), Pitch_ai2(t), Tai2]
/a/	/s/	300	[Pas1(t), Pitch_as1(t), Tas1]
/a/	/s/	500	[Pas2(t), Pitch_as2(t), Tas2]
⋮	⋮	⋮	⋮
/s/	/o/	500	[Pso3(t), Pitch_so3(t), Tso3]

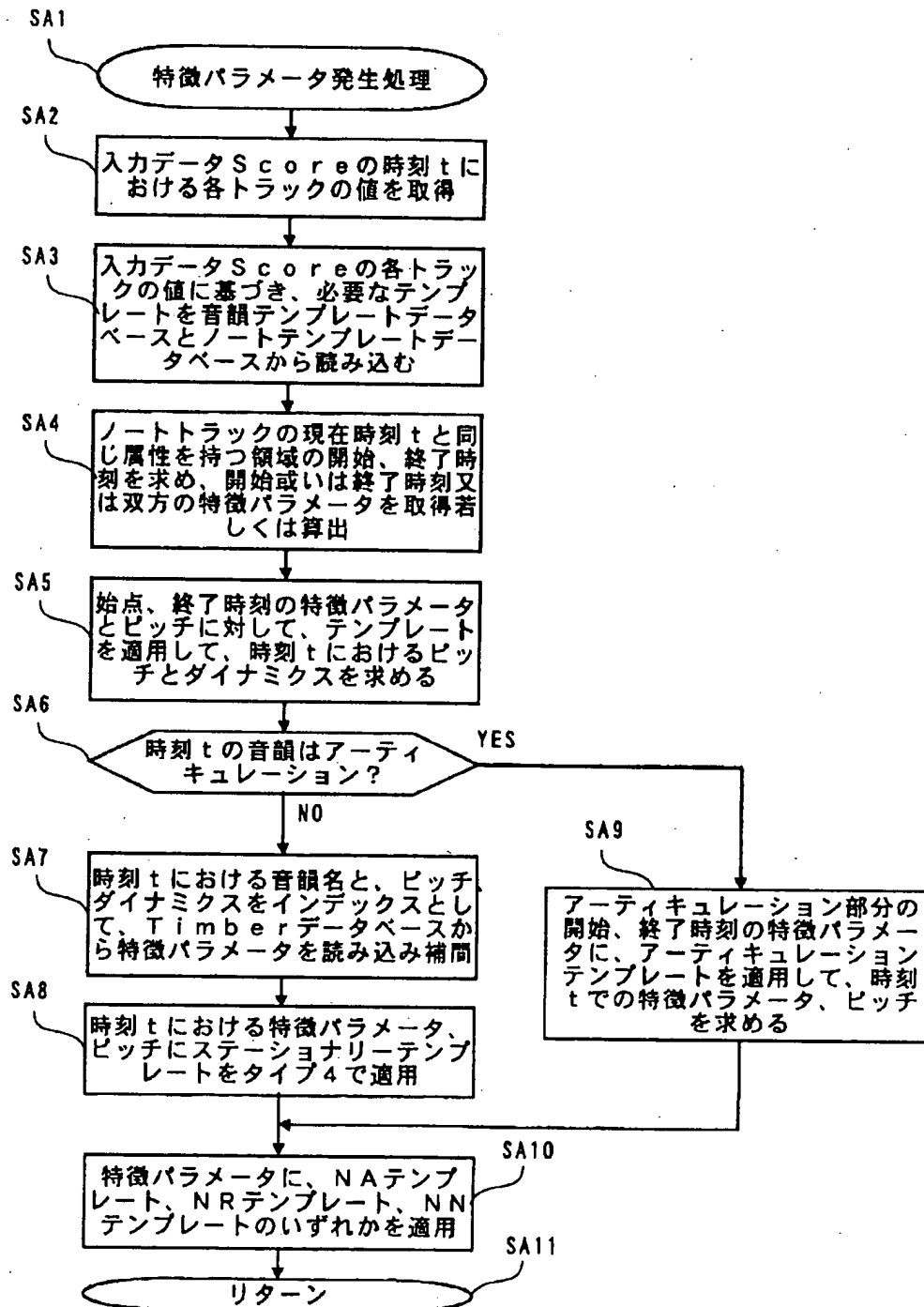
【図7】

音韻名	代表ピッチ[Hz]	特徴パラメータ
/a/	200	[PNA_a1(t), PitchNA_a1(t), TNA_a1]
/a/	300	[PNA_a2(t), PitchNA_a2(t), TNA_a2]
/a/	400	[PNA_a3(t), PitchNA_a3(t), TNA_a3]
/i/	200	[PNA_i1(t), PitchNA_i1(t), TNA_i1]
/i/	300	[PNA_i2(t), PitchNA_i2(t), TNA_i2]
⋮	⋮	⋮
/n/	500	[PNA_n4(t), PitchNA_n4(t), TNA_n4]

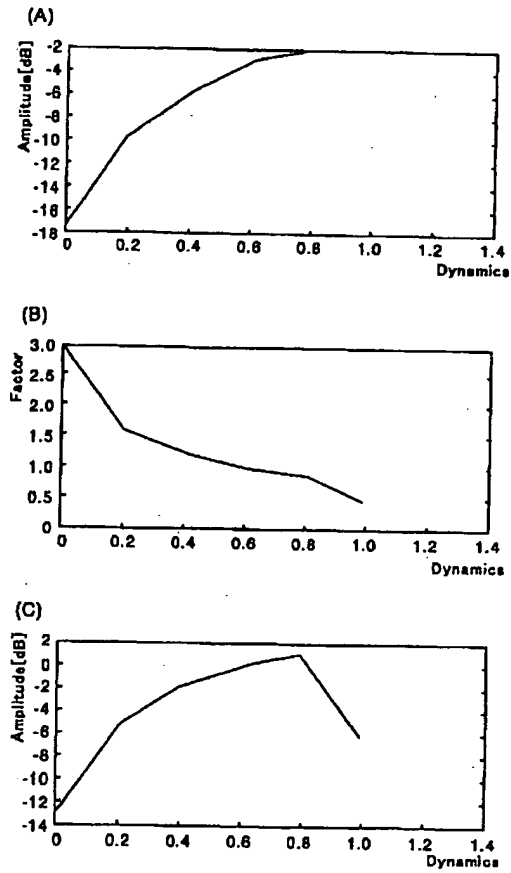
【図8】

音韻名	始点ピッチ[Hz]	終点ピッチ[Hz]	特徴パラメータ
/a/	200	100	[PNN_a1(t), PitchNN_a1(t), TNN_a1]
/a/	200	300	[PNN_a2(t), PitchNN_a2(t), TNN_a2]
/a/	400	300	[PNN_a3(t), PitchNN_a3(t), TNN_a3]
/i/	200	150	[PNN_i1(t), PitchNN_i1(t), TNN_i1]
/i/	300	200	[PNN_i2(t), PitchNN_i2(t), TNN_i2]
⋮	⋮	⋮	⋮
/n/	500	400	[PNN_n4(t), PitchNN_n4(t), TNN_n4]

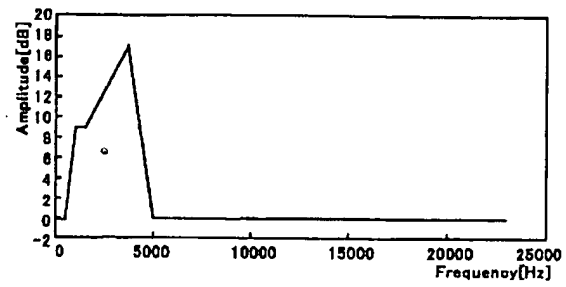
【図9】



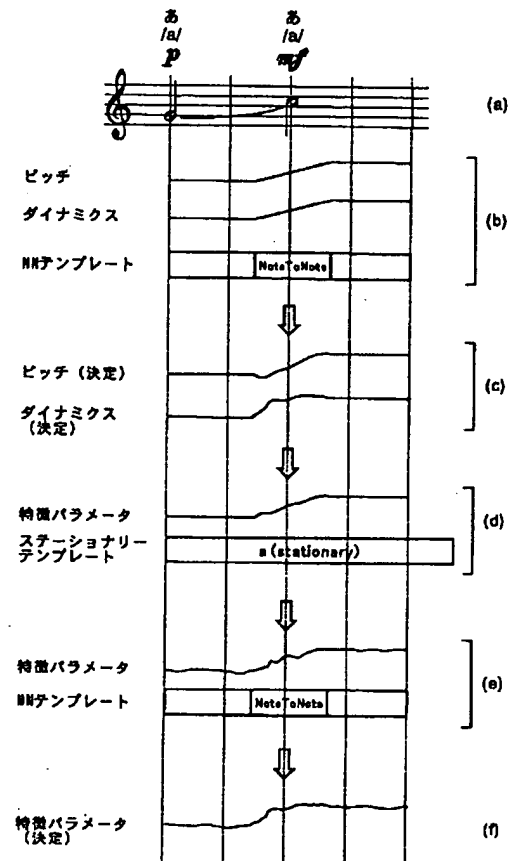
【図10】



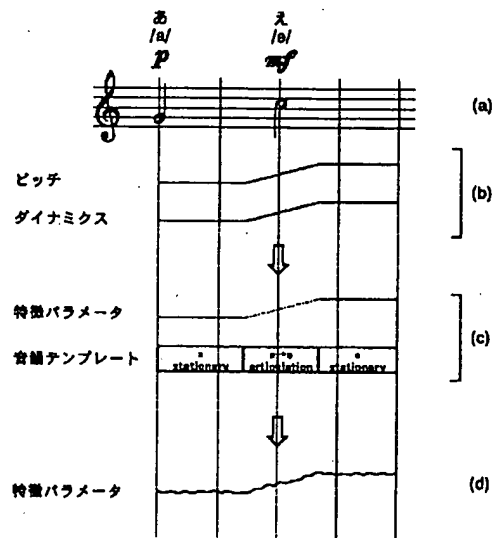
【図11】



【図12】



【図14】



【図15】

